* Modules:

- A module contain 4 basic attribute.
  1) Input & Output
  2) Functions
  3) Mechenics
  4) Internal data.

1) Input & output:

- Input is what it get from invoker while output is what it return to its invoker.

2) Functions:

- Function is what it does to its input to produce its output.

3) Mechenics:

- Mechenics is a procedural code or logic by which it carries out its function.

4) Internal Data:

- It has its own private space i.e. data to which it alone refer.

- Input, output & Function are outside view of module while mechenic & Internal data are inside view of module.

→ Types of Module:

1) Input module
2) Output module
3) Transform module
4) Co-ordinate module
5) Composite module

1) Input module:

subordinate → boss

- It obtain information from ~~there~~ ~~su~~ their subordinate and passes it on to its boss.

2) Output Module: boss → subordinate

- It take information from boss and passes it to its sub-ordinate.

3) Transform module:

- It exist only for ~~set of~~ sack of transforming data into some other form

4) Co-ordinate module:

- It primary concern is managing flow of data to and from different subordinates.

5) Composite Module:

- Sometimes a module can perform more than one task

* Modular Programming :

- It is an approch to programming in which program is broken into several independently compile modules.

- Each module export specify element [such as constant, variable, datatype, Function & Procedure] and all other element remains private to the module

→ Advantages :

1) It is easier and less costly, To add features, modify it or correct the errors after ~~requirements~~ deployment.

2) It is easy to write & debug the program.

3) It is easy to manage, Since more difficult module can be given more skilled programmer & easy module to junier developer.

4) We can divide large complex problem into no. of module, each module manages complexity easily.

5) Modular concept ~~fit~~ fit in well ~~with~~ with top down approch.

→ Disadvantage:

1) It is difficult to learn although principle are clear because there are few formal design technique.

2) Modular programming require more design efforts & care.

3) Modular Programming may slightly more memory space at run time.

4) To avoid slow processing & some OS may have to ensure that modules that call each other frequently are in same machine page